NASA/TM-2017–219629

Implementation of a Battery Health Monitor and Vertical Lift Aircraft Testbed for the Application of an Electrochemistry-based State of Charge Estimator

Timothy R. Potteiger
Vanderbilt University, Nashville, Tennessee

Kenneth W. Eure
Langley Research Center, Hampton, Virginia

David Levenstein
Christopher Newport University, Newport News, Virginia

June 2017

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM-2017–219629

Implementation of a Battery Health Monitor and a Vertical Lift Aircraft Testbed for the Application of an Electrochemistry-based State of Charge Estimator

Timothy R. Potteiger
Vanderbilt University, Nashville, Tennessee

Kenneth W. Eure
Langley Research Center, Hampton, Virginia

David Levenstein
Christopher Newport University, Newport News, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

June 2017

Available from:
NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

Table of Contents

Table of abbreviations and acronyms

ADC	Analog to Digital Converter
AV	Audio Visual
BBB	BeagleBone Black
EM	Electromagnetic
ESC	Electronic Speed Control
I^2C	Inter-Integrated Circuit
PC	Personal Computer
PCB	Printed Circuit Board
PWM	Pulse Width Modulation
SDA	Serial Data Line
SCL	Serial Clock Line
SOC	State Of Charge
SSH	Secured Shell
UART	Universal Asynchronous Receiver/Transmitter
UKF	Unscented Kalman Filter

Implementation of a Battery Health Monitor and Vertical Lift Aircraft Testbed for the Application of an Electrochemistry-based State of Charge Estimator

Abstract - Prediction methods concerning remaining charge in lithium-ion batteries that power Unmanned Aerial Vehicles (UAVs) are of critical concern for the safe fulfillment of mission objectives. In recent years, lithium-ion batteries have been the power source for both fixed wing and vertical lift electric UAVs. The purpose of this document is to describe in detail the implementation of a battery health monitor for estimating the state of charge of a lithium-ion battery and a lithium-ion polymer battery used to power a vertical lift aircraft test-bed. It will be demonstrated that an electro-chemistry based state of charge estimator effectively tracks battery discharge characteristics and may be employed as a useful tool in monitoring battery health.

1. Introduction

Battery state of charge estimation is an area of research that is having its merits continually expanded as eco-friendly systems such as electric and hybrid land, air and sea vehicles continue to gain popularity. Such systems, for the sake of maximizing the utility of the system with respect to battery usage, call for state of charge estimation to be incorporated into a battery health monitor as a co-principal part of the system design. In this paper, we will detail the implementation of a battery health monitor that serves to collect discharge data which is used in processing and implementation of an electric vertical lift aircraft test-bed. The choice to use a test-bed that replicates a vertical lift aircraft with respect to battery discharge is due to the close match between the discharge profile of the test bed and that of the vertical lift aircraft.

The vertical lift aircraft test-bed consists of several parts as shown in Figure 15 and Figure 19. It is comprised of a saw horse with six motors mounted to the sides of the saw horse main beam. Each motor has a propeller attached to it. The motors are commanded by electronic speed controllers and the speed controllers are commanded by an Arduino Uno (a hobbyist grade microcontroller). The motors are powered by a lithium-ion polymer battery that consists of five individual lithium-ion polymer cells. The types of motors, propellers, and electronic speed controllers are consistent with the ones used in a small to medium size vertical lift aircraft to ensure a replication of discharge with the lithium-ion polymer battery. The Arduino controller is programmed to command the motors to spin. The on state involves a uniform motor rotational speed so a constant discharge of the lithium-ion polymer battery occurs. A constant discharge corresponds to the discharge that the vertical lift aircraft would experience if it simply were to hover.

The basic hardware implementation of the battery health monitor consists of an ADC, a current sensor, a Beaglebone Black, an opto-isolated modules, a custom made PCB, and an I^2C cable. This experiment is designed to be performed offline using battery voltage and current measurements as estimator input data. The estimated state consist of internal battery charge values with known initial values. A connector between the battery and the discharge circuit has a current sensor along with voltage measurement.

The battery health monitor operates at 27 Hz and is powered from an independent source. This implies that at 27 Hz, the Beaglebone Black sends a request to sample to the ADC, then receives the samples for the current and voltage from the ADC, and stores the samples using dynamic memory allocation. The state of charge estimation is performed offline on the gathered data using an unscented Kalman filter which incorporates an electrochemistry-based model. The unscented Kalman filter was chosen for its efficiency and its ability to track highly nonlinear trajectories well as demonstrated with the reentry problem [1]. The electrochemistry based model

is chosen as it has been developed and successfully demonstrated on multiple platforms [2], [3], [4], [5].

This paper is organized as follows. Section 2 presents the electrochemistry-based battery model. Section 3 describes the unscented Kalman filter used for state of charge estimation. Section 4 describes the hardware setup. Results are presented in Section 5 and conclusions are drawn in Section 6. An appendix is given to supply additional information to the interested reader concerning the derivation of the unscented Kalman filter.

2. Electrochemistry-based Model

This section describes an electrochemistry model of the lithium-ion battery. The aspects of the electrochemistry model that allow it to capture the nonlinear capacity effects of a lithium-ion battery are noted. Finally, the dynamics of a lithium-ion battery during discharge and charge are described and the mathematical model is defined.

2.1 Model Differentiation

The electrochemistry based models are a part of a set of three well noted battery state of charge modeling techniques which also include mathematical based models and electrical models [6]. Mathematical models include methods such as regression and artificial neural networks which are not derived explicitly on physical mechanisms that take place within the battery during discharge. These models instead are fit to training data, which can be a drawback. For instance, it would be a very daunting task to gather training data for a neural network that expressed the state transition from every possible state given every possible input. Electrical models, unlike mathematical models, attempt to capture the chemical processes of the battery. They are known as empirical models. Devising an electrical model is done by constructing a circuit with components chosen to match battery behavior. These models are typically more computationally efficient than electrochemistry-based models but are known to suffer the drawback of not being able to capture the nonlinear capacity effects of lithium-ion batteries. Electrochemistry models are more accurate than empirical models. However, they come with a high computational cost that ranges per the method employed [7].

What sets the electrochemistry-based model used in this paper and previously presented in [2] apart from other electrochemistry-based models is that it is an ohmic porous-electrode model that offers the benefit of being computationally efficient through the usage of a set of ordinary differential equations to model the update of the charges at the electrodes. The model has been shown to capture the nonlinear capacity effects. Such effects consist of the rate capacity effect and the recovery effect [8]. The impact of the rate capacity effect on voltage is described by the drop or rise in voltage due to the change in current. The model takes this effect into account with the use of an ohmic resistance that causes the voltage to subside at higher currents.

The impact of the recovery effect on voltage is described as the recovery of voltage during periods of rest. This allows, if there is a period of rest, for the battery to recover charge and extend the discharge time, where the discharge time does not include the period of rest. The ability of the electrochemistry-based model to capture the recovery effect comes with the involvement of intra-electrode diffusion in the charge update equations. Diffusion within the electrodes takes place between the surface and bulk layers allowing the negative electrode to recover charge and increase battery voltage.

2.2 Electrochemistry Model Outline

The basic kinetics for the electrochemistry model are derived from the chemical reactions that take place at the electrodes and are shown in equations 1 through 4 [2]. Equations 1 and 2 are the reactions at the electrodes during a discharge. During discharge, an oxidation reaction takes

place at the negative electrode resulting in lithium-ions and electrons. The electrons travel from the surface of the negative electrode to the surface of the positive electrode leaving the surface potential of the negative electrode increased and the surface potential of the positive electrode decreased causing them to converge towards the same potential. This means that the voltage will decrease as the voltage across the battery is the difference between the negative electrode surface potential and the positive electrode surface potential. The lithium-ions travel from the bulk of the negative electrode to the bulk of the positive electrode through the electrolyte. A reduction reaction occurs at the positive electrode when lithium-ions diffuse into the surface and the electrons diffuse into the bulk. The result is the increase in the surface potential at the positive electrode which counters the decrease from the initial introduction of the electrons.

$$Li_xC \rightarrow C + xLi^+ + xe^- \tag{1}$$

$$Li_{1-x}CoO_2 + xLi^+ + xe^- \rightarrow LiCoO_2 \tag{2}$$

Equations 3 and 4 are the reactions that take place at the electrodes during charge. During charge, oxidation occurs at the positive electrode and results in lithium-ions and electrons as products. The electrons travel from the surface of the positive electrode to the surface of the negative electrode resulting in an increase in the surface potential at the positive electrode and a decrease in the surface potential at the negative electrode. The surface potentials diverge during charge which increases the voltage. The lithium-ions travel from the bulk of the positive electrode to the bulk of the negative electrode through the electrolyte. A reduction reaction takes place at the negative electrode when lithium-ions diffuse into the surface and electrons diffuse into the bulk. This results in an increase in the surface potential at the negative electrode which counters the decrease in potential from the introduction of the electrons that traversed from the positive electrode surface.

$$C + xLi^+ + xe^- \rightarrow Li_xC \tag{3}$$

$$LiCoO_2 \rightarrow Li_{1-x}CoO_2 + xLi^+ + xe^- \tag{4}$$

To define the electrochemistry-based model, the relevant voltages and how they impact the voltage of the battery must be detailed. First, the battery voltage that the model is capturing and our system is measuring is seen in Figure 1 [2] to be the difference in potential between the surfaces of the negative and positive electrodes.

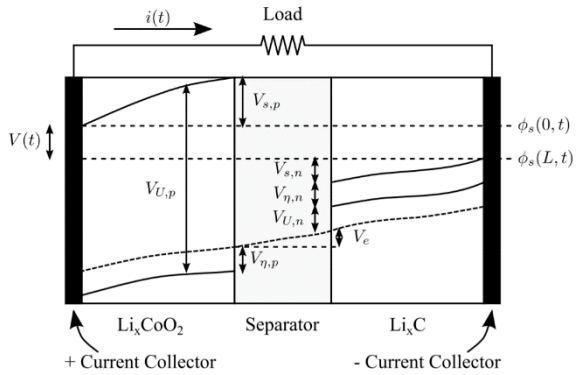


Figure 1: Diagram of Relevant Battery Voltages

The voltages that factor into the determination of the battery voltage can be stated in relation to how they detract from the ideal voltage. This voltage is defined in accordance to Figure 1 as the difference between $V_{U,p}$ and $V_{U,n}$. The first set of voltages that detract from the equilibrium

voltage are known as ohmic voltage drops. The ohmic voltages are comprised of the electrolyte ohmic voltage denoted V_e, the solid phase ohmic voltage of the positive electrode denoted $V_{s,p}$, and the solid phase ohmic voltage of the negative electrode denoted $V_{s,n}$. The second set of voltages that detract from the equilibrium voltage are known as surface overpotentials and are a result of charge transfer resistance and solid electrolyte interface kinetics [2]. The surface overpotentials are comprised of a negative electrode surface overpotential denoted $V_{\eta,p}$ and a positive electrode surface overpotential denoted $V_{\eta,n}$.

Now that the voltages are defined, the state definition, the input, the output, and the relevant model variables are defined in equations 5 through 7 [5]. The state vector is defined in equation 5 and consists of the positive electrode surface charge $q_{s,p}$, the positive electrode bulk charge $q_{b,p}$, the negative electrode bulk charge $q_{b,n}$, the negative electrode surface charge $q_{s,n}$, the sum of the ohmic voltage contributions V_o', the positive electrode overpotential $V_{\eta,p}'$, and the negative electrode overpotential $V_{\eta,n}'$. The input vector is defined in equation 6 and consists of the discharge current i_{app}. The output vector is defined in equation 7 and consists of the battery voltage V.

$$x(t) = [\, q_{s,p}\ q_{b,p}\ q_{b,n}\ q_{s,n}\ V_o'\ V_{\eta,p}'\ V_{\eta,n}' \,] \tag{5}$$

$$u(t) = [\, i_{app} \,] \tag{6}$$

$$y(t) = [\, V \,] \tag{7}$$

The state transition equations for the electrode charges incorporate the discharge current and diffusion, as shown in equations 8 through 12 [2]. Inter-electrode electron flow occurs primarily at the surface of the electrodes. Hence, the current does not have a direct impact on charge at the bulk of the electrodes. Indirectly, the current changes the concentration gradient of lithium-ions between the surface and the bulk of the electrodes which impacts diffusion and in turn the charge at the bulk of the electrodes. The cumulative charge equations are shown in equations 13 through 15 [2].

$$\dot{q}_{s,p} = i_{app} + \dot{q}_{bs,p} \tag{8}$$

$$\dot{q}_{b,p} = -\dot{q}_{bs,p} \tag{9}$$

$$\dot{q}_{b,n} = -\dot{q}_{bs,n} \tag{10}$$

$$\dot{q}_{s,n} = -i_{app} + \dot{q}_{bs,n} \tag{11}$$

$$\dot{q}_{bs,i} = \frac{1}{D}(c_{b,i} - c_{s,i}) \tag{12}$$

$$q_p = q_{s,p} + q_{b,p} \tag{13}$$

$$q_n = q_{s,n} + q_{b,n} \tag{14}$$

$$q_{max} = q_p + q_n \tag{15}$$

These charge equations are coupled directly with the concentration and lithium mole fraction equations shown in equations 16 through 20 [2]. In equations 16 and 17, v refers to volume and in equations 16 through 20, the subscript i refers to either electrode. The lithium-ion mole fraction for the positive electrode must be at least 0.4 and the lithium-ion mole fraction at the negative electrode must be no more than 0.6. If the positive electrode has a mole fraction of lithium less than 0.4, lithium cannot be reversibly removed [9]. The mole fractions at the positive electrode

and negative electrode are complimentary so together they must add up to 1. Hence, the mole fraction at the negative electrode must be no greater than 0.6.

$$c_{b.i} = \frac{q_{b.i}}{v_{b.i}} \tag{16}$$

$$c_{s,i} = \frac{q_{s,i}}{v_{s,i}} \tag{17}$$

$$x_i = \frac{q_i}{q_{max}} \tag{18}$$

$$x_{s,i} = \frac{q_{s,i}}{q_{s,i,max}} \tag{19}$$

$$x_{b,i} = \frac{q_{b,i}}{q_{b,i,max}} \tag{20}$$

The equations to solve for the cumulative ohmic voltage, the surface overpotentials, and the intermediate variables are shown in equations 21 through 29 [2] [5]. In equation 21, U_0 is the reference potential. In equations 21 through 29, R is the universal gas constant, T is the electrode temperature, n is the number of electrons transferred in from the chemical reaction in equation 1, and F is Faraday's constant. R_o is the cumulative resistance from the ohmic voltages drop, α is the symmetry factor, and S_i is the area of the electrode. The terms $\tau_{\eta,p}$, and $\tau_{\eta,n}$ denote time constants. Lastly, $A_{i,k}$ are parameters for the Redlich-Kister expansion. The Redlich-Kister expansion is used to capture the activity coefficient terms related to excess Gibbs free energy [9]. For a more detailed outline of the intermediate variables see [2].

$$V_{u,i} = U_{0,i} + \frac{RT}{nF}\ln\left(\frac{1 - x_{s,i}}{x_{s,i}}\right) + V_{INT,i} \tag{21}$$

$$V_{INT,i} = \frac{1}{nF}\sum_{k=0}^{N_i} A_{i,k}((2x_{s,i} - 1)^{k+1} - \frac{2x_{s,i}k(1 - x_{s,i})}{(2x_{s,i} - 1)^{1-k}}) \tag{22}$$

$$V_o = i_{app}R_o \tag{23}$$

$$V_{\eta,i} = \frac{RT}{F\alpha}\text{arcsinh}(\frac{J_i}{2J_{i0}}) \tag{24}$$

$$J_i = \frac{i_{app}}{S_i} \tag{25}$$

$$J_{i0} = k_i(1 - x_{s,i})^{\alpha}(x_{s,i})^{1-\alpha} \tag{26}$$

$$\dot{V_o}' = \frac{V_o - V_o'}{\tau_o} \tag{27}$$

$$\dot{V}_{\eta,p}' = \frac{V_{\eta,p} - V_{\eta,p}'}{\tau_{\eta,p}} \tag{28}$$

$$\dot{V}_{\eta,n}' = \frac{V_{\eta,n} - V_{\eta,n}'}{\tau_{\eta,n}} \tag{29}$$

The equation to solve for the battery voltage is shown in equation 30 and the state of charge equations are shown in equations 31 and 32 [2]. The nominal state of charge refers to the percent of charge left at the negative electrode while the apparent state of charge refers to the percent of charge left at the surface of the negative electrode. If the charge at the surface of the negative electrode becomes entirely depleted, discharge cannot continue until charge diffuses from the bulk of the negative electrode so long as the nominal state of charge has not been entirely depleted in that instance. The constants for the electrochemistry-based model are shown in Table 3. The Redlich-Kister coefficients for the lithium-ion battery are shown in Table 1. The Redlich-Kister coefficients for the lithium-ion polymer battery are shown in Table 2.

$$V = V_{u,p} - V_{u,n} - V_o' - V_{\eta,p}' - V_{\eta,n}' \tag{30}$$

$$SOC_{nominal} = \frac{q_n}{0.6q_{max}} \tag{31}$$

$$SOC_{apparent} = \frac{q_{s,n}}{0.6q_{s,n,max}} \tag{32}$$

Table 1: Lithium-ion Battery Model Parameters

Parameter	Value
$U_{0,p}$	4 V
$A_{p,0}$	-38246 J/mol
$A_{p,1}$	-12651 J/mol
$A_{p,2}$	23624 J/mol
$A_{p,3}$	-78405 J/mol
$A_{p,4}$	1122.5 J/mol
$A_{p,5}$	321970 J/mol
$A_{p,6}$	84521 J/mol
$A_{p,7}$	-1075200 J/mol
$A_{p,8}$	1656 J/mol
$A_{p,9}$	989930 J/mol
$A_{p,10}$	282570 J/mol
$A_{p,11}$	-161280 J/mol
$A_{p,12}$	-507580 J/mol
$U_{0,n}$	0.01 V
$A_{n,0}$	86.19 J/mol
q_{max}	15,750 C
R_0	0.2524 Ω
D	4.6831x10^6 mol s/C/m^3

Table 2: Lithium-ion Polymer Battery Model Parameters

Parameter	Value
$U_{0,p}$	4.16 V
$A_{p,0}$	-17343 J/mol
$A_{p,1}$	8523.8 J/mol
$A_{p,2}$	2953.1 J/mol
$A_{p,3}$	-440 J/mol
$A_{p,4}$	-178 J/mol
$A_{p,5}$	-310 J/mol
$A_{p,6}$	634 J/mol
$A_{p,7}$	-394 J/mol
$A_{p,8}$	-1900 J/mol
$A_{p,9}$	-464 J/mol
$A_{p,10}$	-511 J/mol
$A_{p,11}$	8021.7 J/mol
$A_{p,12}$	-9000 J/mol
$U_{0,n}$	0.01 V
$A_{n,0}$	86.19 J/mol
q_{max}	40170 C
R_0	0.0045 Ω
D	4.2157x10^6 mol s/C/m^3

Table 3: Battery Model Constants

Parameter	Value
R	8.3144 J/mol/K
T	292.1 K
F	96487 C/mol
n	1
τ_0	2.08671 s
α	0.5
S_p	3.0962x10^{-4} m^2

k_p	248898 A/m^2
$v_{s,p}$	2x10^{-6} m^3
$v_{b,p}$	1.8x10^{-5} m^3
$\tau_{n,p}$	46.4311 s
S_n	4.3755x10^{-4} m^2
k_n	2120.96 A/m^2
$v_{s,n}$	2x10^{-6} M^3
$v_{b,n}$	1.8x10^{-5} M^3
$\tau_{\eta,n}$	1001.38 s

3. Unscented Kalman Filter

Given the nonlinear model of Section 2.2, a method is needed to estimate the state defined in equation 5 given the battery current i_{app} and the output voltage measurement *V*. The Unscented Kalman Filter (UKF) is used here for nonlinear state estimation. The equations are given below and can be found in reference [1]. Appendix A gives the reader background on the UKF. First, the weights *w* and sigma points χ are calculated. Here n_x is the system order, κ is a tuning constant, $\hat{x}$ is the mean, and *P* is the state autocorrelation. The superscript *i* is the *i*th column. It is very important that the electro-chemistry model be initialized properly for the given battery. Table 1, Table 2, and Table 3 give the values used in this implementation.

$$w^0 = \frac{\kappa}{n_x + \kappa} \tag{33}$$

$$w^i = \frac{1}{2(n_x + \kappa)}, \qquad i = 1,2,\dots,2n_x \tag{34}$$

$$\chi^0 = \hat{x}_{k-1|k-1} \tag{35}$$

$$\chi^i = \hat{x}_{k-1|k-1} + \left(\sqrt{(n_x + \kappa)P_{k-1|k-1}}\right)^i, \qquad i = 1,2,\dots,n_x \tag{36}$$

$$\chi^i = \hat{x}_{k-1|k-1} - \left(\sqrt{(n_x + \kappa)P_{k-1|k-1}}\right)^i, \qquad i = n_x + 1, n_x + 2,\dots,2n_x \tag{37}$$

The prediction step involves passing the sigma points through the nonlinear system model. The state equation is $f(\cdot)$ and the measurement equation is $g(\cdot)$. Both the state and measurement equations are determined from the electro-chemical model of Section 2.2. The input *u* is the battery current i_{app} and *k* is the current time step.

$$\hat{\chi}^i = f\left(\chi^i, u_{k-1}\right), \qquad i = 1,2,\dots,n_s \tag{38}$$

$$\hat{Y}^i = g\left(\chi^i\right), \qquad i = 1,2,\dots,n_s \tag{39}$$

$$\hat{x} = \sum_i^{n_s} w^i \hat{\chi}^i \tag{40}$$

$$\hat{y}_{k|k-1} = \sum_{i}^{n_s} w^i \hat{Y}^i \tag{41}$$

$$P_{k|k-1} = Q + \sum_{i}^{n_s} w^i \left(\hat{\chi}^i - \hat{x}_{k|k-1}\right)\left(\hat{\chi}^i - \hat{x}_{k|k-1}\right)^T \tag{42}$$

The update steps are shown below; n_s is the number of sigma points. The notation $k|k-1$ means the current time step *k* given measurement data up to time $k-1$. The update occurs by using the new voltage measurement $y_k = V$ to update the state estimate.

$$P_{yy} = R + \sum_{i}^{n_s} w^i \left(\hat{Y}^i - \hat{y}_{k|k-1}\right)\left(\hat{Y}^i - \hat{y}_{k|k-1}\right)^T \tag{43}$$

$$P_{xy} = \sum_{i}^{n_s} w^i \left(\hat{\chi}^i - \hat{x}_{k|k-1}\right)\left(\hat{Y}^i - \hat{y}_{k|k-1}\right)^T \tag{44}$$

$$K_k = P_{xy} P_{yy}^{-1} \tag{45}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(y_k - \hat{y}_{k|k-1}\right) \tag{46}$$

$$P_{k|k} = P_{k|k-1} - K_k P_{yy} K_k^T \tag{47}$$

In the above algorithm, *Q* is the process noise covariance matrix and *R* is the measurement noise covariance matrix.

4. Hardware and Software Setup

The hardware setup is designed to gather data from a lithium-ion battery and lithium-ion polymer battery during discharge. The data is post processed by the unscented Kalman filter using the electro-chemistry battery model. The first section of the hardware and software setup details the hardware used, the hardware connections, and the software control flow for the battery health monitor. With the battery health monitor, discharge data can be gathered from a lithium-ion battery. The second section of the hardware setup details the construction of the vertical lift aircraft testbed, how it is controlled, and how it is interfaced with the battery health monitor to gather discharge data from a lithium-ion polymer battery.

4.1 Battery Health Monitor

The connections between the battery health monitor components are shown in Figure 2 and Figure 3. The primary controller is the Beaglebone Black. The Beaglebone is chosen as the microcontroller because it has a 1 GHz Arm cortex A8 processor with a Debian Linux operating system, 2x46 pin headers, and a small lightweight frame. The data gathering module is a Linear Technology LTC2309 12-bit ADC that has a 5-volt reference voltage. The ADC is powered by a 5-volt source as opposed to a 3.3-volt source that powers the Beaglebone. This requirement adds an extra degree of complexity to the design as the Beaglebone must not only power but interface the ADC as well. The ADC I^2C pins operate from a 0 to 5-volt range and the Beaglebone I^2C pins operate from a 0 to 3.3-volt range. The difference in operating voltages between the Beaglebone and the ADC presents an inability to communicate over I^2C with a direct connection of the SDA, SCL, VCC, and GND pins of each device. Communication is achieved between the two devices using an Adafruit Bidirectional level shifter. The function of the level shifter is to step-up the SDA

and SCL voltages of the Beaglebone from 3.3 volts to 5 volts when the Beaglebone is sending data to the ADC and to step-down the SDA and SCL voltages of the ADC from 5 volts to 3.3 volts when the ADC is sending data to the Beaglebone. For the data gathering, the ADC takes samples of the battery voltage and the discharge current using a 100-amp current sensor. The design requirements for this battery health monitor mandate that the current sensor has a 9-volt reference voltage. The output of the current sensor is 5 to 0 volts representing 0 to 100 amps and 5 to 10 volts representing 0 to -100 amps. The reference voltage to the current sensor is provided by a Pololu S18V20ALV adjustable step-up/step-down voltage regulator that steps up 5 volts from the Beaglebone to 9 volts. To ensure the reference voltage to the ADC remains constant and does not vary due to a change in the power source to the Beaglebone, a Pololu D24V22F5 step-down voltage regulator is placed between the aforementioned step-up/step-down regulator and the ADC to provide a constant 5-volt source to the ADC. For safety and noise reduction purposes, input opto-isolated modules are used to isolate the battery and current sensor from the ADC. A Dataforth 8B31-02 module is used to isolate the current sensor from the ADC for a current reading and a Dataforth 8B31-09 module is used to isolate the battery from the ADC for a voltage reading. For convenience, a custom PCB board is used that embeds the input opto-isolated modules, the level shifter, and ADC into one unit. The battery tested is a 2200 mAh, 3.7 V nominal, 18650 non-protected rechargeable TLIFE lithium-ion battery.

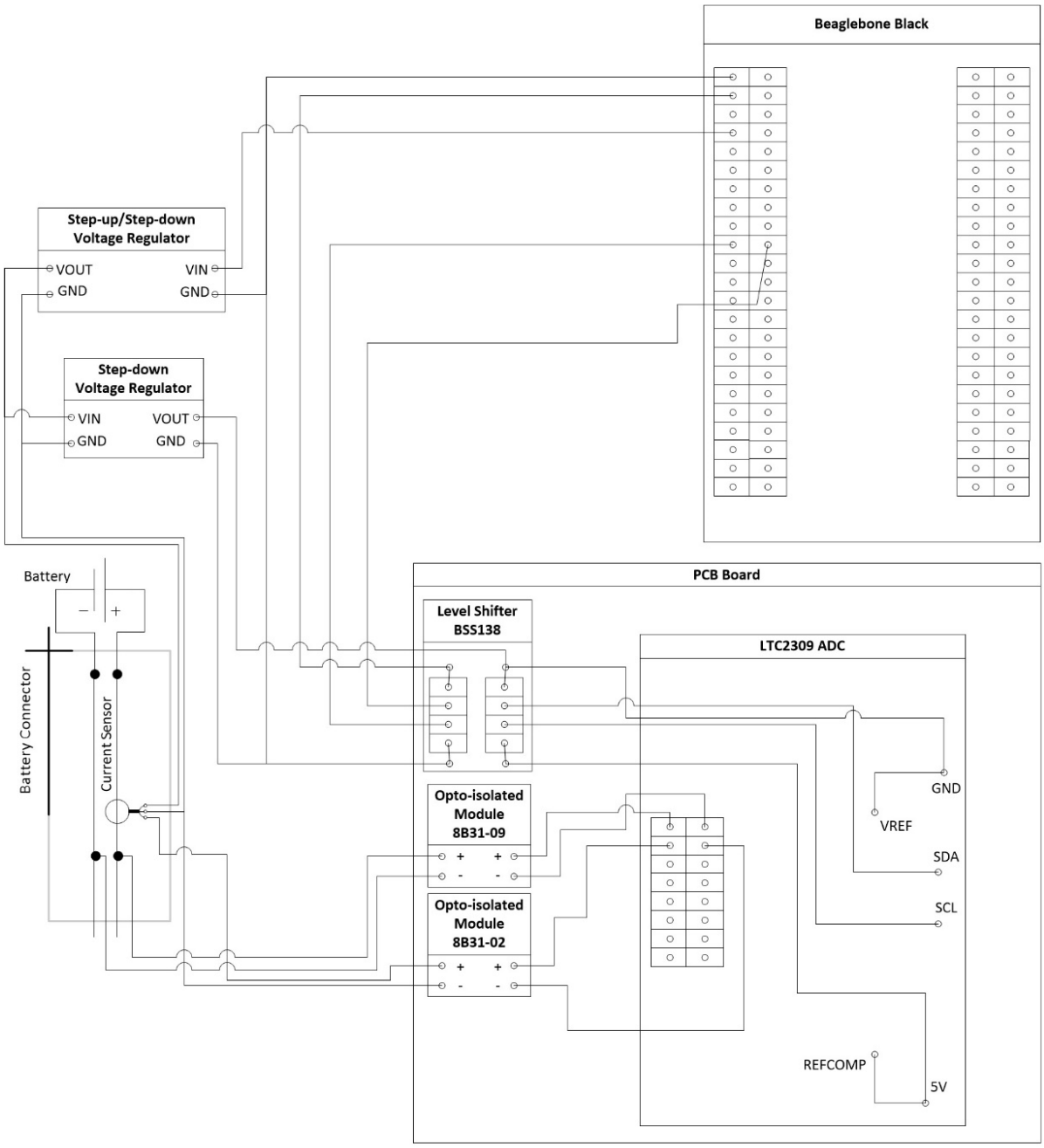


Figure 2: Battery health monitor schematic

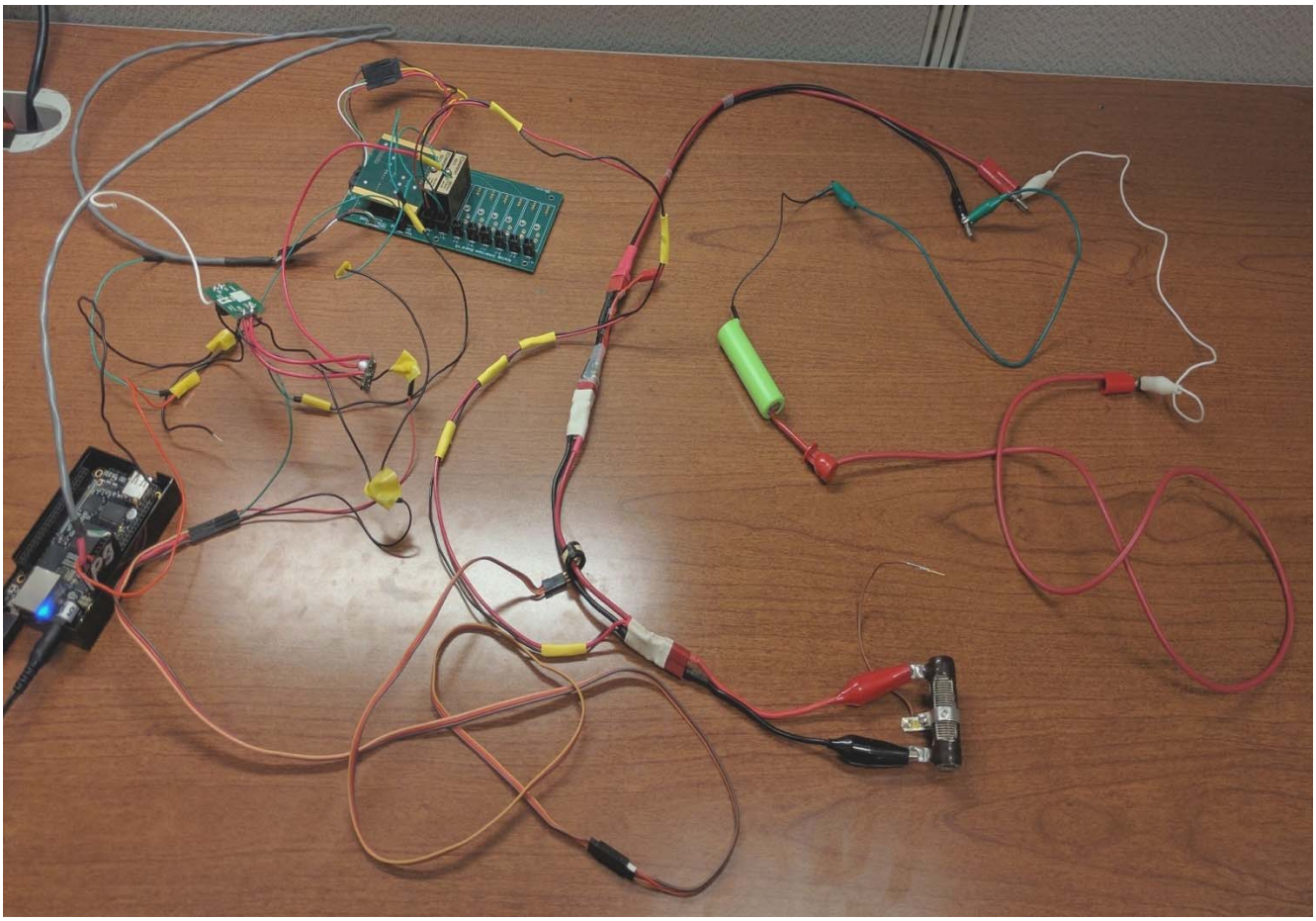
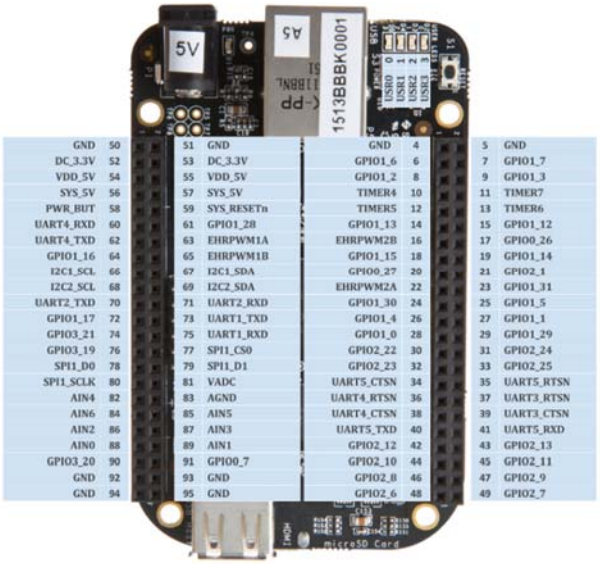


Figure 3: Battery health monitor setup for lithium-ion battery

Detailing the connections between the Beaglebone and the ADC starts with outlining the I²C pins on both. I²C communication requires four lines. They are VDD, GND, and SDA for data transmission, and SCL for clock. Using Figure 4 to indicate which pins on the Beaglebone are involved in the I²C protocol, we note that pin ***52*** (***DC_3.3V***) is used for VDD, pin ***50*** (***GND***) is used for GND, pin ***69*** (***I²C2_SDA***) is used for SDA, and pin ***68*** (***I²C2_SCL***) is used for SCL. Figure 5 indicates which pins on the ADC are involved in the I²C protocol. To bridge the pins involved in the I²C protocol on the Beaglebone with the pins involved in the I²C protocol on the ADC, a level shifter, a step-down voltage regulator, and a step-up/step-down voltage regulator are used.



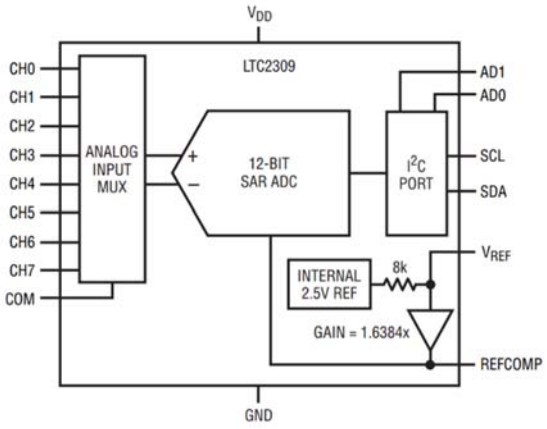


Figure 4: Beaglebone Black pin map

Figure 5: LTC2309 ADC pin map

Using Figure 4 through Figure 8, we describe the connections between the Beaglebone, the ADC, the level shifter, the step-up/step-down voltage regulator, and the step-down voltage regulator as shown in Figure 9. Note that before including the step-up/step-down regulator in the circuit, we tuned its voltage adjustment potentiometer to ensure that ***VOUT*** was 9 volts.

Figure 6: Adafruit level shifter

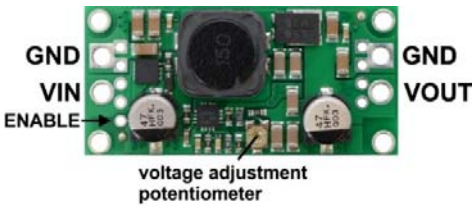


Figure 7: Step-up/step-down voltage regulator

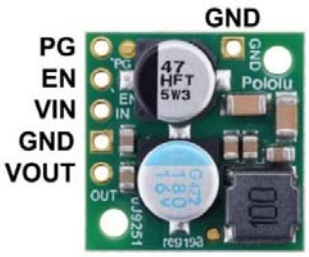


Figure 8 : S tep-down voltage regulator

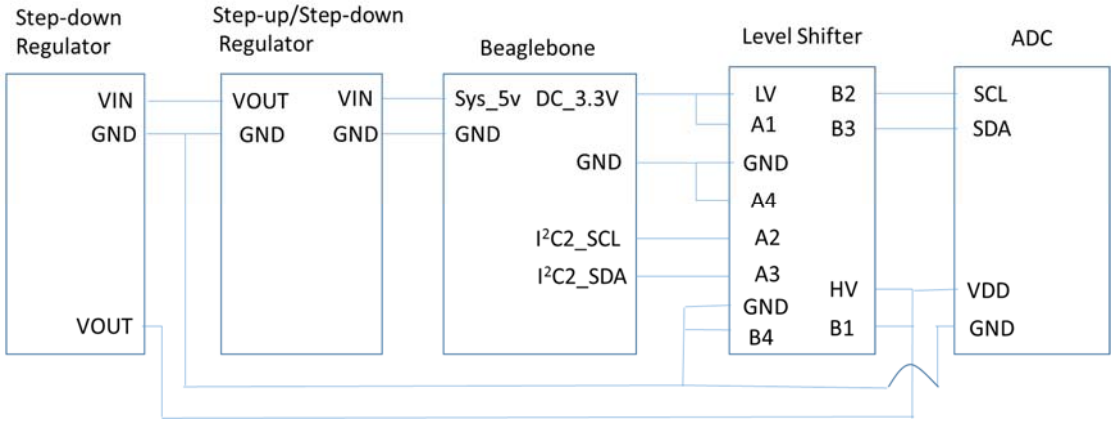


Figure 9: Connection diagram for battery monitor

The PCB board, with the top level shown in Figure 11, is responsible for routing the connections between the level shifter and the ADC along with the connections between the opto-isolated modules outputs and the ADC. The PCB board has a header that allows the I^2C connections from the Beaglebone to be made. The I^2C header routes to the level shifter pins. The opto-isolated module inputs are routed to headers on the PCB board. A special connector is made to fit between the battery and discharge circuit that stems a parallel branch to connect to the PCB board header for the Channel 0 opto-isolated module input and has a current sensor that is attached on the "hot" line of the connector which outputs a signal to the PCB board header for the Channel 1 opto-isolated module input. The current sensor is given a reference voltage of 9 volts from the step-up/step-down regulator. Using Figure 7 and Figure 10 as references to the pin mappings of the two devices, the connections between the current sensor and the step-up/step-down regulator are the following. ***VOUT*** from the step-up/step-down maps to the current sensor pin denoted 1. ***GND*** from the step-up/step-down regulator maps to the middle current sensor pin denoted 0. The far right current sensor pin denoted 0 is the output voltage with a level that is indicative of the current. The current sensor output voltage and ***GND*** map to the PCB board header for the Channel 1 opto-isolated module input. The opto-isolated modules serve a few other important functions aside from isolating the discharge circuit from the battery health monitor. The chief among those functions include ensuring a maximum voltage threshold delivered to the ADC and stepping down the input voltage. The 8B31-02 module outputs voltages between 5 and -5 volts and accepts input between 5 and -5 volts. When the input is greater than 5 volts, the output

voltage saturates at 5 volts and when the input is less than -5 volts, the voltage saturates at -5 volts. As the 8B31-02 is mounted on Channel 1 which is connected to the current sensor, the specification of the interaction between the two devices must be clarified. The current sensor can output a voltage in the range of 0 to 9 volts. The module saturates at 5 volts. This means with the interaction between the current sensor and the module, we lose the ability to track voltages from 5 to 9 volts. The range of 5 to 9 volts represents -11 to -100 amps with a negative polarity referring to a charging condition. These charging conditions should not occur for preserving battery life. Therefore, for this application, we neglect the voltage range of 5 to 9 volts from the current sensor. This decision gives a benefit of a higher precision in our current measurement. The difference in amps represented by two consecutive quantized levels captured by the 12-bit ADC is 0.027 amps. The 8B31-09 module outputs voltages between 5 and -5 volts and accepts input between 40 and -40 volts. This module is sufficient for mounting on Channel 0 and receiving the voltage of the battery from the parallel branch of the connector as the expected voltage of the lithium-ion polymer batteries used in the experiment are nominally 18.5 volts. The output to input ratio is 8 which implies a step-down factor of 8. The difference in volts represented by two consecutive quantized levels captured by the 12 ADC is 0.00977 volts. This means that the voltage measurements are more precise than our current measurements.

Figure 10: Current sensor

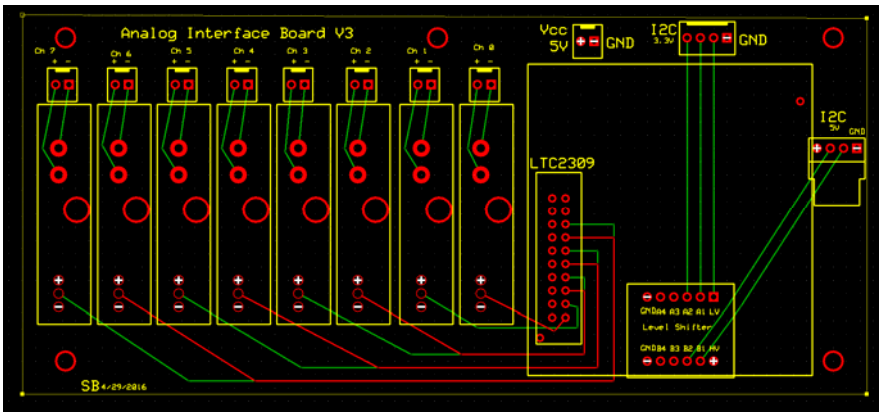


Figure 11: Custom PCB board outline

The control flow of the software starts with the battery health monitor code. The main code is written in C for the Beaglebone. The main code is structured to sample the current and voltage 27 times a second, filter the samples, store the samples using dynamic memory allocation, print out the instantaneous samples once a second, and end when the voltage is depressed below a lower threshold. The 27 Hz sample rate is accomplished through polling the time until the time is equal to or is after the ideal sampling time. When this condition is satisfied, the samples are taken and the ideal sampling time becomes the last ideal sampling time plus the sampling period. The ideal sampling times are independent of whether the previous samples were taken precisely at the ideal sampling times. This method ensures the average sampling period converges to 0.037037 seconds (27Hz). For instance, if the program samples half a period after the ideal sampling period, the next iteration will ideally sample a half a period after the previous sample was taken. This allows the shorter sampling period to cancel out with the previous elongated sampling period and average with the previous sampling periods to result in approximately the ideal sample period. The sampling itself is done through the VoltageRead and CurrentRead functions written in C language. It is important to mention the programmatic format of the functions. In the VoltageRead and CurrentRead functions, the read ADC function is called twice. When the ADC is referenced through the read ADC function the first time, the previous sample is returned and the ADC is told to sample the address specific channel. When the ADC is referenced through the read ADC function the second time, the sample that the previous read ADC function

initiated is returned. This is the sample that is returned by the VoltageRead and CurrentRead functions. The address used in the VoltageRead function to reference Channel 0 is 0x88 and the address used in the CurrentRead function to reference Channel 1 is 0xC8 [10]. The filtering of the samples is done with a basic 1 pole infinite impulse Butterworth filter. The filter has a -3dB cutoff frequency of 3Hz and is designed for a sampling rate of 27 Hz. The frequency response is shown in Figure 12 and Figure 13.

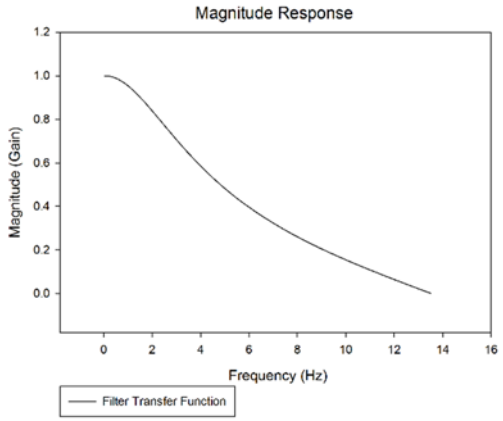


Figure 12: Magnitude Response of Filter

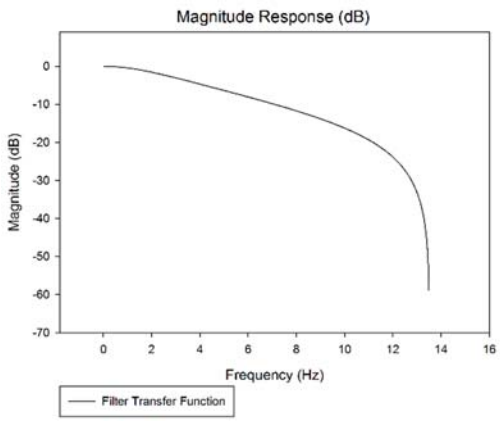


Figure 13: Magnitude Response (dB) of Filter

To achieve the function of storing the samples using dynamic memory allocation, two linked lists are incorporated: one for the voltages and one for the currents. These lists are singly-linked and consist of several nodes that point to each other. Each node consists of a pointer to the next node and a value of type double. Each linked list consists of a first node called root, a final node called tail, and a size value of type int. A global node is declared but not initialized so that it can be used as a 'null' value to signal the end of the list. This global value is never altered. To add a value to the list, the addNode function is called. This function takes in parameters for the linked list, which includes the value to be added to this list. When entering this function, the size of the list is checked to be 0 or 1. If the size is 0, the root node is allocated in memory, set to hold the value given in the parameters, and set to point to the null node. If the size is 1, a new node is declared and allocated in memory to hold the value given with the parameters. It is also set to point to the null node. The root node is set to point to this new node and the tail node is set to equal this new node. If the size of the list is neither 0 nor 1, the list must have at least two nodes. This case requires a new node to be allocated in memory and initialized in the same way the previous case does. The differences are that the root node does not need to be altered while the tail node does. The tail node is set to point to the node that was just created. It is then set to equal this new node. The size of the list is incremented by 1 every time this function is called.

The printFile function is called to print the lists to a file. This function takes in a voltages linked list and a currents linked list. Two nodes are created but not allocated in memory: one is set to equal the root node of the voltages list, and the other is set to equal the root node of the currents list. These are used so that the list can be iterated through without changing any node values. To start the algorithm in this function, the size of the lists must be at least 1. A while loop is used to ensure that the two new nodes are not pointing to the null node (since the lists must have the same size, only one of the lists needs to be checked). Inside this loop, both values of the nodes are printed to the file. The nodes are then set to equal what they each are pointing to. In other words, the nodes become the next nodes in the lists. Once the nodes are pointing to the null node, the while loop terminates. However, the values of the final nodes have not yet been printed. Thus, these values are printed to the file outside of the while loop. After every five voltage and current values, the function prints to a new line of the file. For every loop iteration of the code, the addNode function is called twice: once for the instant voltage and once for the instant current.

The function of printing out the voltage and current once every second is accomplished by keeping track of a counter, incrementing it every iteration, printing out the voltage and current when the counter reaches 27, and setting the counter back to 1. These instant voltages are used to stop the main loop when the voltage decreases below a lower threshold. The printFile function is called outside of the main loop to print the voltages and currents to a file just before the program terminates.

The data file is then transferred from the Beaglebone for offline processing which consists of fitting the electrochemistry-based model and running the unscented Kalman filter with the data. The fitting of data is performed with the Nelder-Mead search algorithm. The Nelder-Mead algorithm is a simplex method that searches for input values of a function that result in a local minimum of the function [11]. The function minimized using the search algorithm is the square difference between the electrochemistry-based model and discharge data. The unscented Kalman filter that is used to process the captured data is a part of the PrognosticsAlgorithm library and the electrochemistry-based model is a part of the PrognosticsModel library [12] [13]. To allow for an overloaded Battery.Create function a constructor for Parameters is made that allows the maximum charge, cumulative ohmic resistance, diffusion coefficient, and Redlich-Kister coefficients as inputs and adjusts the default parameters accordingly. The sample rate in Parameters is also adjusted to have a default value of 0.037037 seconds (27 Hz) for sampleTime. The unscented Kalman filter is then used with the parameters found from the Nelder-Mead algorithm.

4.2 Vertical Lift Aircraft Testbed

The vertical lift aircraft testbed is designed to be used with a lithium-ion polymer battery to replicate the discharge of the battery on a hexacopter. The testbed setup consists of three distinct systems: the testbed, the testbed controller, and the battery health monitor. A saw pony (small saw horse) is chosen as the testbed since it has a body that it is long enough to have 3 motors mounted on each side and gives enough space between the motors to allow for non-interfering operation when the propellers are attached. The motors are mounted on the base of the saw pony with three on each side. The motors are KDE-Direct 4215XF-465 three phase brushless motors. They are used in the hexacopter shown in Figure 14 and are designed for heavy-lift electric multi-rotor applications. After the motors are mounted, electronic speed controllers are connected to the three phase connections of the motors. The electronic speed controllers are KDE-Direct XF-UAS 55HVC. They can supply 55+ amps to the motors and can operate at high voltages. The ESCs have three wires that are outputs to the three phase motors, three wires that are inputs which take in a PWM signal, and two leads that connect to the battery. The leads on all the ESCs are soldered together and attached to a connector to the battery. The battery is a Thunder Power RC ProLite+Power 25C 7800 mAh 5-cell/5s 18.5V LiPo battery. The wires between the electronic speed controllers and the motors are tied down on the saw pony using zip ties to ensure they do not interfere with the propellers. The three wires on the ESCs that are dedicated to PWM signals are connected to three wire extension cables that are connected to the controller. The ESCs are programmable and have been assigned a range of PWM values between 1100 and 1900 microseconds to represent minimum and maximum motor speed. For debugging purposes, when the battery is connected to the ESCs, they beep periodically. The ESCs stop beeping once they are supplied with a PWM value. For safety reasons, the initial PWM value supplied to the ESCs should be less than the minimum PWM value required to start the motors. The propellers used are KDE-Direct CF155-TP propeller blades and are 15.5 inches in diameter and a pitch of 5.3 inches. The constructed testbed is shown in Figure 15.

Figure 14: Hexacopter

Figure 15: Saw pony hexacopter testbed

The controller for the saw pony testbed is implemented with an Arduino Uno. The hardware setup of the controller also involves a switch, a resistor, and an Xbee-Pro 900 DigiMesh RF Module. The connection between the switch, resistor, and Arduino are described as follows. The bottom pin of the switch is attached to a pull-up resistor which connects to pin ***7*** of the Arduino. The bottom pin of the switch also connects to pin ***A0*** of the Arduino. The top pin of the switch connects to ***GND*** of the Arduino. Pin ***7*** of the Arduino is set to high. When the switch is open, ***A0*** is read as a high value and when the switch is closed, ***A0*** is read as a low value. The connection between the Arduino and the Xbee module is the following. Pin ***2***, which is assigned to be ***Tx*** in the controller software of the Arduino, is attached to the ***Dout*** pin of the Xbee. Pin ***12***, which is assigned to be ***Rx*** in the controller software of the Arduino, is attached to the ***Din*** pin of the Xbee. The ***3.3V*** pin of the Arduino is attached to the ***Vcc*** pin of the Xbee and the ***GND*** pin of the Arduino is attached to the ***GND*** pin of the Xbee. The interface to the PWM cables of ESCs from the Arduino is built through extending wires from the pins of the Arduino to connectors. The connectors have 3 holes for pins crimped onto wire to be inserted. They are for ***GND***, ***Vcc***, and ***Sig*** wires. The connections from the Arduino are the following. The ***GND*** pin of the Arduino is connected to a wire that branches off into six wires which are crimped with male pins and inserted into the ***GND*** slot in all the connectors. The ***5V*** pin of the Arduino is connected to a wire that branches off into six wires as well which are crimped with male pins and inserted into the ***Vcc*** slot in all the connectors. Pins ***3***, ***5***, ***6***, ***9***, ***10***, and ***11*** of the Arduino all connect to wires that are crimped with male pins and inserted into the ***Sig*** slot in the connectors. The interconnections for the controller are shown in Figure 16.

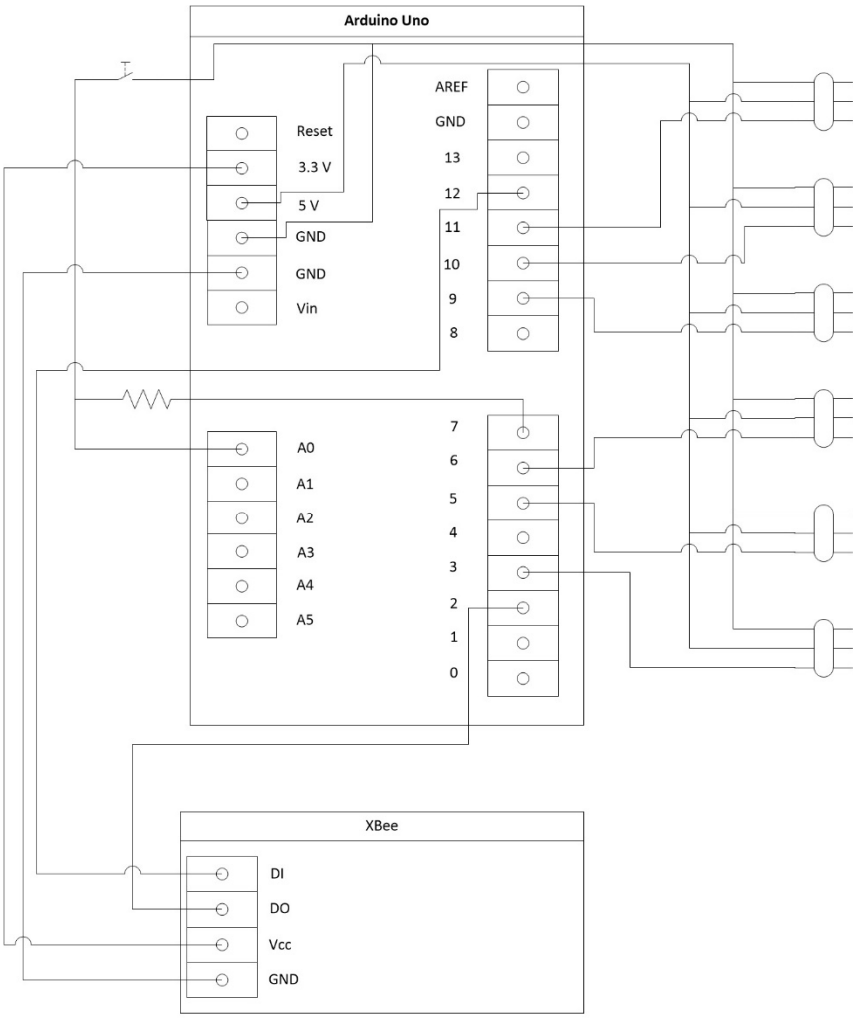


Figure 16: Arduino Uno saw pony controller schematic

The controller software can be described by a Moore state machine. The two states are Go and Stop with the default state being Stop. The output of the Stop state is a PWM value of 800 microseconds being sent to the motor ESCs. This is the output in the Stop state as it is intended to suppress the ESC beeping through providing a PWM value that is recognized by the ESCs but is not high enough to cause the motors to spin. The output of the Go state is a PWM value that is higher than 1100 microseconds to cause the motors to spin. A transition occurs every 500 milliseconds based on serial input and the value of the switch. The guard condition for a transition from the Stop state to the Go state is that the sampled value of ***A0*** must be above 120, which means the switch is open, and the serial input must be a ‘g’. If this condition is not satisfied, the Stop state transitions back to itself. The guard condition from the Go state to the Stop state is that the sampled value of ***A0*** must be below 120, which means the switch is closed, or the serial input must be a ‘s’. If this condition is not satisfied, the Go state transitions back to itself.

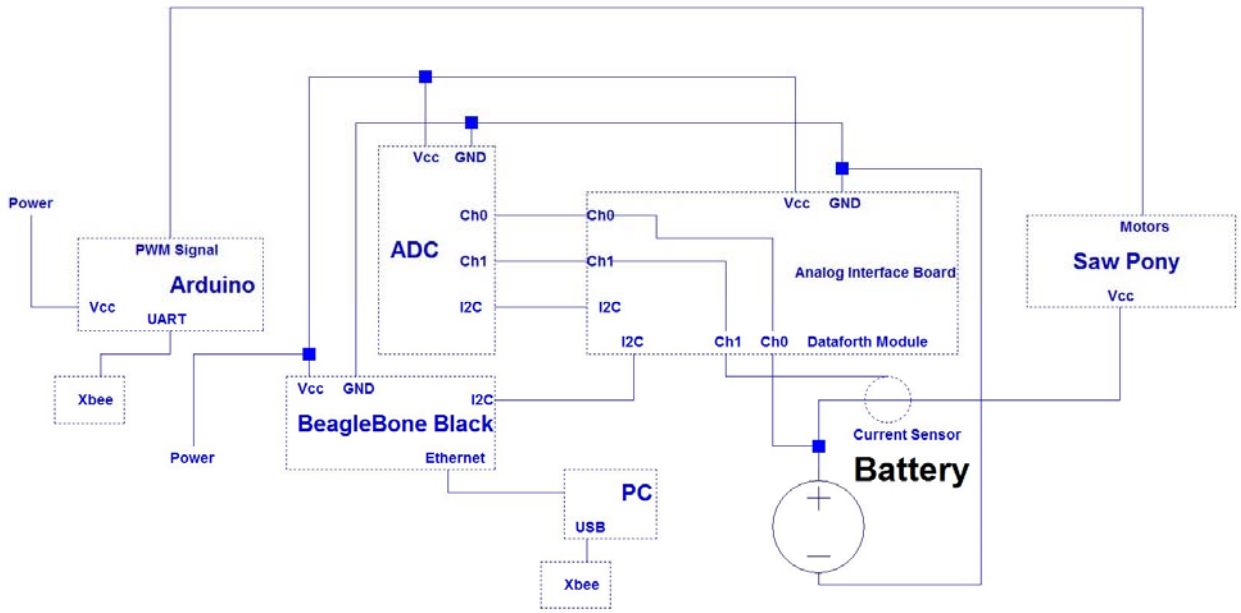


Figure 17: Saw pony testbed block diagram

The block diagram of the test setup is shown in Figure 17. A computer is used to communicate with the Beaglebone through an Ethernet connection. The PC controls the Beaglebone through a SSH connection. While running the battery health monitor code, the SSH terminal displays the voltage and current every second to give the user an idea of how close the battery is to crossing the lower voltage threshold and closing the program. The Beaglebone receives voltages and currents from the battery through a custom PCB Analog Interface Board via the ADC. Since the Analog Interface Board and ADC can only take in voltages, the current sensor converts the current into a voltage between 0 and 9 volts. The Analog Interface Board has two Dataforth Modules, which convert higher voltages (above 5 volts) to lower voltages within the ADC sample range. The ADC converts this analog voltage to a digital voltage and sends it back to the Analog Interface Board through an I^2C connection. The Analog Interface Board sends this data to the Beaglebone through a separate I^2C port. The current is translated from a voltage reading into an Ampere reading in the battery health monitor code using calibration data. Both the Analog Interface Board and the ADC receive power through the Beaglebone, which receives its power from a 120-volt AC source.

The computer is also used to send signals from an Xbee connected through USB to another Xbee connected to the Arduino through UART. Commands are sent through a serial terminal. The Arduino sends a PWM signal to the motors on the saw pony, which is used to turn the motors on and off. These motors are powered by the battery. The PC controls this signal through a serial terminal. The Arduino receives power from a separate 120-volt AC source. The Arduino does not directly communicate to the Beaglebone which means when the battery health monitor code stops running, the Arduino is not prompted by the Beaglebone to turn off the motors. This is an open loop system which means that the user is required to provide input to the system to turn off the motors when it is noticed that the battery health monitor code has finished or the battery voltage has depleted below a set level.

Figure 18: Test chamber

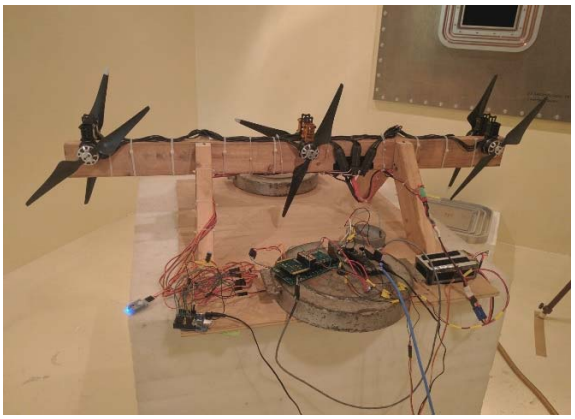


Figure 19: Saw pony testbed with Arduino controller and battery health monitor

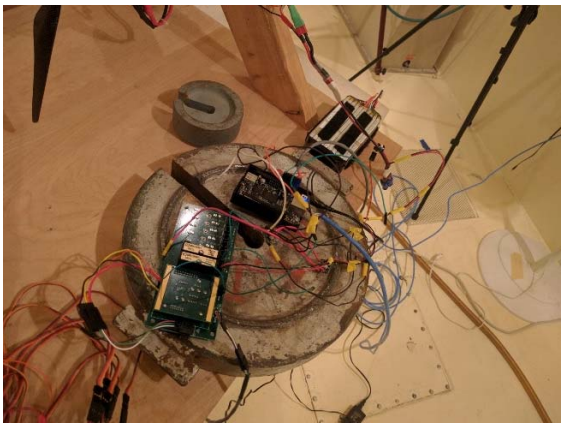


Figure 20: Battery health monitor connection to saw pony testbed

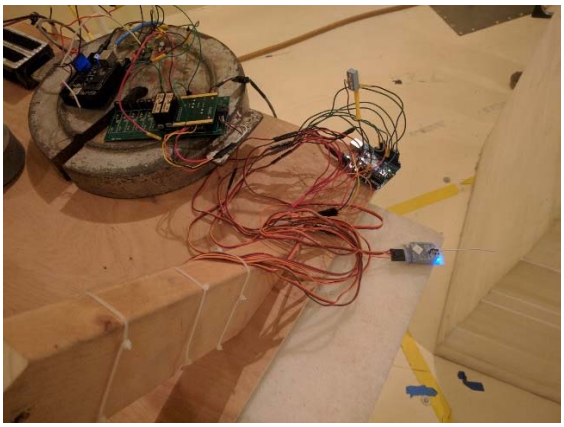


Figure 21: Arduino controller connection to saw pony testbed

Figure 22: Operation of saw pony testbed

Figure 23: Observation chamber

The saw pony, saw pony controller, and battery health monitor are placed in a test chamber shown in Figure 18 through Figure 21 [14]. The operation of the saw pony testbed is shown in Figure 22. The computer is placed in an observation chamber shown in Figure 23. An EM wave relay is connected between the test chamber and the observation chamber that can receive an EM signal on one side of the relay and emitting the EM signal on the other side of the relay. This allows the Xbee in the test chamber that is connected to the Arduino to receive data from the Xbee in the observation chamber that is connected to the computer and vice versa. An Ethernet cable is routed between the test chamber and the observation chamber. This allows for the SSH connection between the Beaglebone and the computer. For extra feedback to the user, an AV cable was routed between the test chamber and the observation chamber. In the test chamber, a camera is mounted to a stand that is facing towards the saw pony and is connected to one side of the video feed cable. In the observation chamber, the other side of the video feed is connected to a monitor and displays the video that the camera is capturing. Once the test has finished, the data is transferred from the Beaglebone for offline processing which is detailed in the Battery Health Monitor subsection.

5. Results

This section presents the gathered discharge data and the analysis results using the battery health monitor data. The first discharge data set is collected by having a 1 ohm resistor load connected to the lithium-ion battery. The data is filtered with the MATLAB smooth function. The filtered data is then used with the Nelder-Mead search algorithm to find a set of parameters for the electrochemistry-based model that provides a reasonable fit to the filtered data. The comparison between the filtered data and the electrochemistry-based model that uses the parameters in Table 1 and Table 3 is displayed in Figure 24.

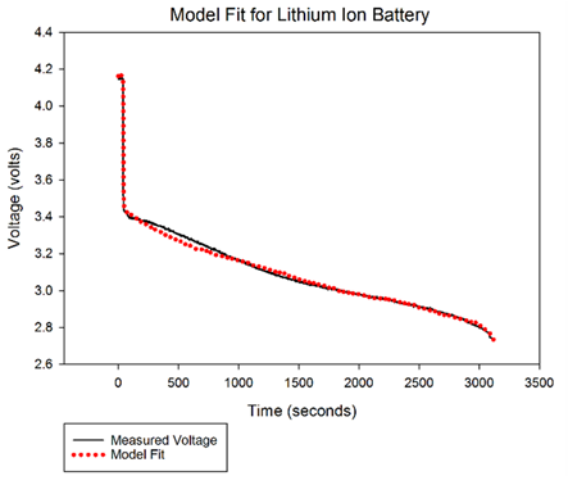


Figure 24: Electrochemistry model fit compared to lithium-ion battery discharge

The first discharge data set is then used for offline processing with the unscented Kalman filter. Figure 25 and Figure 26 show the raw data for the voltage and current measurements taken using the Beaglebone. It should be noted that the raw data has been put through the infinite impulse response filter implemented on the Beaglebone. The voltage output of the unscented Kalman filter is displayed in Figure 27. The nominal and apparent state of charge calculations using the updated state estimate of the charges are displayed in Figure 28. As an indication of correct function, the state of

charge is shown to converge to depletion at the same time the voltage approaches its knee.

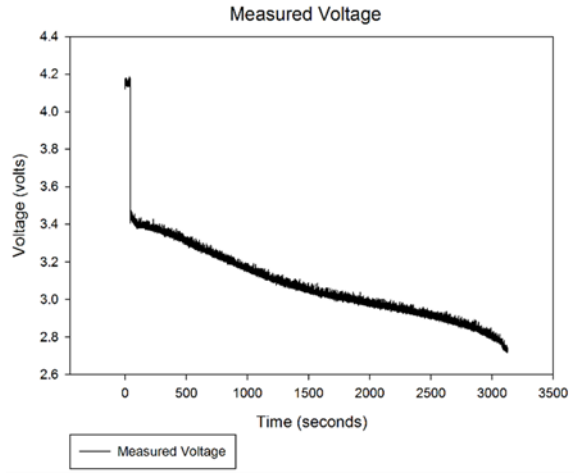


Figure 25: Lithium-ion battery constant discharge measured voltage

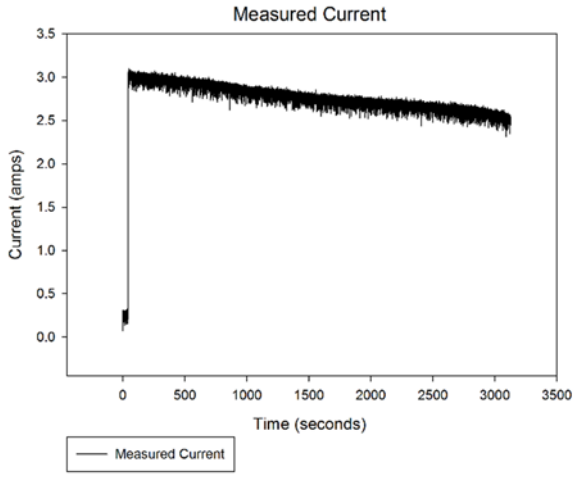


Figure 26: Lithium-ion battery constant discharge measured current

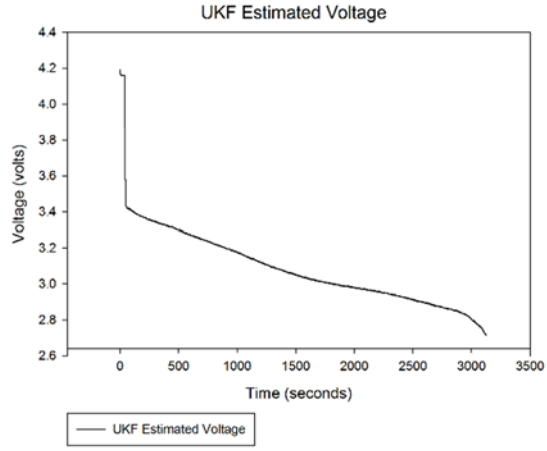


Figure 27: Lithium-ion battery constant discharge UKF estimated voltage

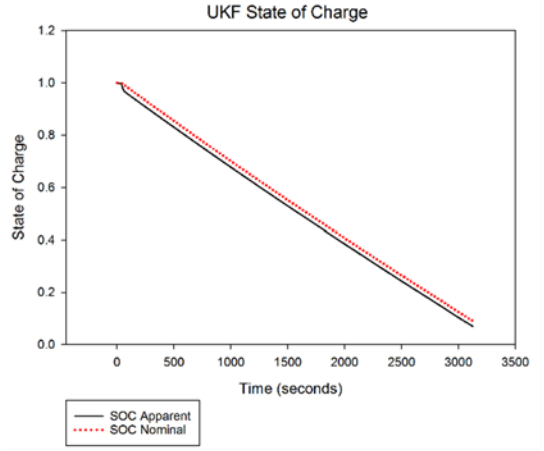


Figure 28: Lithium-ion battery constant discharge UKF state of charge estimation

The second set of discharge data is collected under the same setup using the 1 ohm resistor. The difference in the discharge from the first scenario is that the battery is sporadically disconnected from the resistor to provide an intermittent discharge. The raw data for the voltage and current measurements is shown in Figure 29 and Figure 30. The output of the unscented Kalman filter for the voltage is shown in Figure 31. The nominal and apparent state of charge calculations using the updated state estimate of the charges are displayed in Figure 32. As with the first data set, the fact that the state of charge is shown to converge to depletion at the same time the voltage approaches its knee is an indication of correct function.

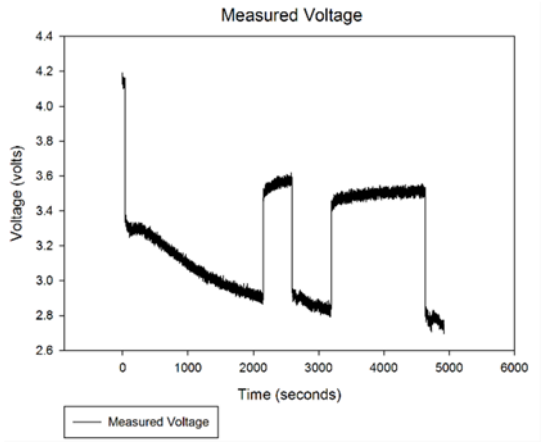


Figure 29: Lithium-ion battery intermittent discharge measured voltage

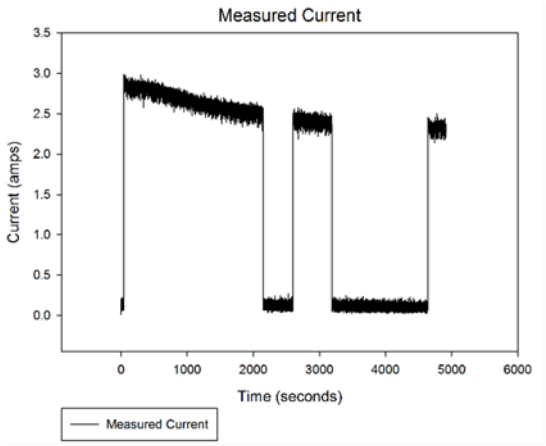


Figure 30: Lithium-ion battery intermittent discharge measured current

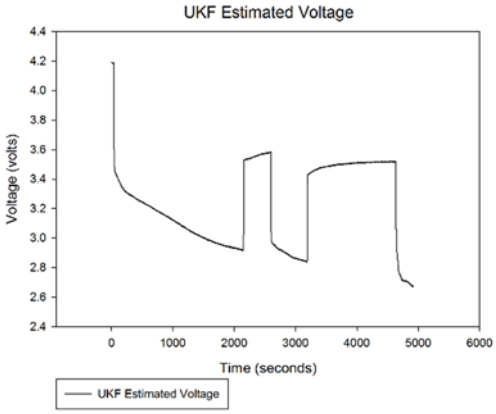


Figure 31: Lithium-ion battery intermittent discharge UKF estimated voltage

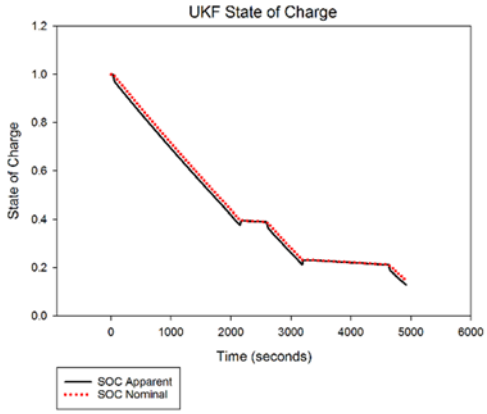


Figure 32: Lithium-ion battery intermittent discharge UKF state of charge estimation

The third discharge data set is collected by having the saw pony testbed connected to the lithium-ion polymer battery. The lithium-ion polymer battery consists of 5 lithium-ion polymer battery cells so the assumption is that the cells are equivalent in voltage and the voltage data is divided by 5. The division by 5 is done to compare the voltage of one cell to the electrochemistry-based model as the electrochemistry-based model is designed for a single cell. The data is then filtered with the MATLAB smooth function. The filtered data is used with the Nelder-Mead search algorithm to find a set of parameters for the electrochemistry-based model that provides a reasonable fit to the filtered data. The comparison between the filtered data for a single cell and the electrochemistry-based model that uses the parameters in Table 2 and Table 3 is displayed in Figure 33.

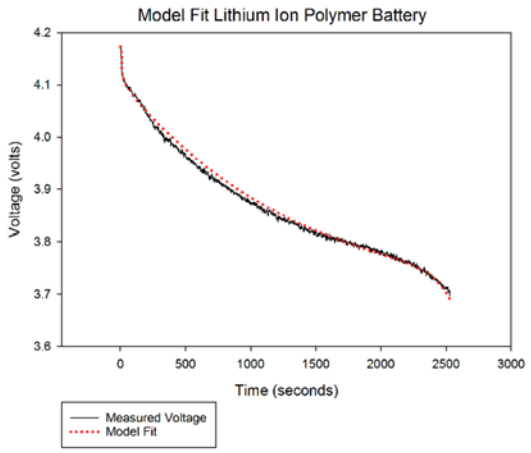


Figure 33: Electrochemistry model fit compared to lithium-ion polymer battery discharge

The third discharge data set is then used for offline processing with the unscented Kalman filter. Figure 34 and Figure 35 show the raw data for the voltage and current measurements using the Beaglebone. It should be noted that the raw data has been put through the infinite impulse response filter implemented on the Beaglebone. The output of the unscented Kalman filter is produced by dividing the raw voltage measurements by 5 to use as input for the unscented Kalman filter and multiplying the output by 5 to get the estimated voltage for the entire 5 cell battery. The estimate for the battery voltage is displayed in Figure 36. The nominal and apparent state of charge calculations using the updated state estimate of the charges are displayed in Figure 37. The convergence of the state of charge to depletion at the same time the voltage approaches its knee indicates correct function.

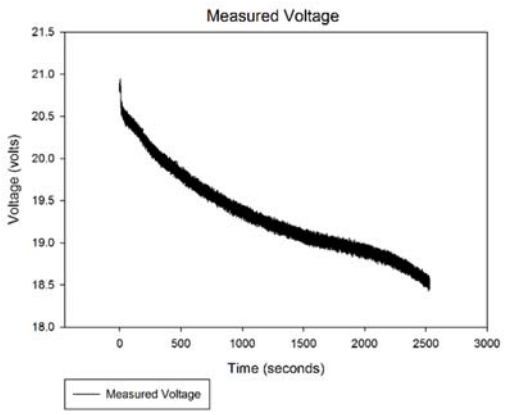


Figure 34: Lithium-ion polymer battery constant discharge measured voltage

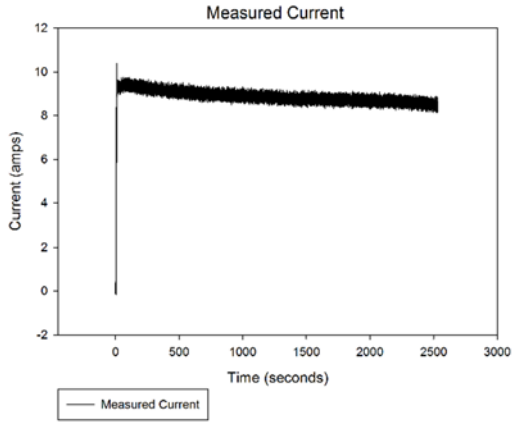


Figure 35: Lithium-ion polymer battery constant discharge measured current

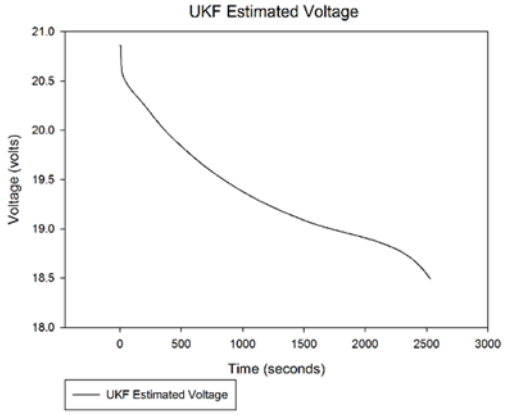


Figure 36: Lithium-ion polymer battery constant discharge UKF estimated voltage

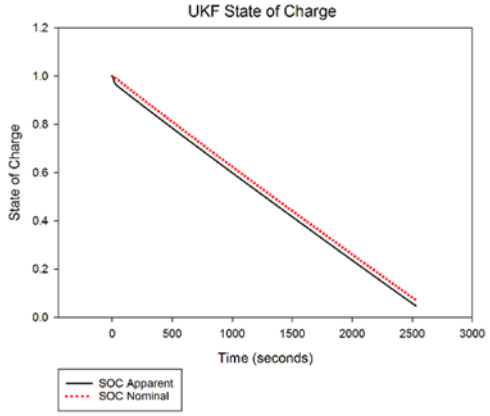


Figure 37: Lithium-ion polymer battery constant discharge UKF state of charge estimation

6. Conclusion

As the progression of Lithium-ion battery modeling and state of charge estimation takes place, it is important to be able to convert these methods into tractable algorithms that can be realized in real-time closed loop systems to help ensure the safe fulfillment of mission objectives. With that in mind, the impact of this work is two-fold. First, the unscented Kalman filter was shown to correctly estimate the state of charge from the data gathered by the battery health monitor. Second, the vertical lift aircraft testbed was shown to power vertical lift components used in a hexacopter from a lithium-ion polymer battery that is used in the hexacopter.

The ability of the unscented Kalman filter to track state of charge using data gathered by the battery health monitor is important for the following reasons. The first reason is that the implementation of the battery health monitor contributes towards the hardware framework of an online implementation of the electrochemistry-based state of charge estimation algorithm. The success of fitting the model used in the UKF to the lithium-ion battery demonstrates its validity and can be compared to model fits in past literature [2]. The second reason was that the model could fit the lithium-ion polymer battery data and estimate its state of charge. This shows that it is valid to assume that the cells in the battery are in equivalent states. The last reason is that the battery health monitor can be used as a hardware framework for a future online implementation

of the unscented Kalman filter. The primary benefit the testbed serves is to gather discharge data from a battery that is used in a vertical lift aircraft without needing to take on the risks of flying.

In conclusion, state of charge estimation is demonstrated in the results section to estimate the state of charge effectively with respect to observations. First, the state of charge approaches 0 when the battery voltage begins to drop off. Second, the shape of the state of charge curve is correct as the state of charge curve should be linear since the discharge current is constant. Also, as expected the apparent state of charge is always less than or equal to the nominal state of charge. Finally, the charge state estimates are noise filtered. This can be inferred from noise being eradicated from the UKF estimated voltage curve as the estimated voltage is calculated using the charge state estimates.

References

[1] S. J. Julier and J. K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, 1997.

[2] M. Daigle and C. S. Kulkarni, "Electrochemistry-based Battery Modeling for Prognostics," in *Annual Conference of the Prognostics and Health Management Society*, New Orleans, LA, 2013.

[3] M. Daigle and C. S. Kulkarni, "A Battery Health Montoring Framework for Planetary Rovers," in *Aerospace Conference*, 2014.

[4] B. Bole, C. S. Kulkarni and M. Daigle, "Adaptation of an Electrochemisty-based Li-ion Battery Model to Account for Deteriorization Observed under Randomized Use," in *Annual Conference of the Prognostics and Health Management Society*, 2014.

[5] M. Daigle, "End-of-discharge and End-of-life Prediction in Lithium-ion Batteries with Electrochemistry-based Aging Models," in *American Institute of Aeronautics and Astronautics*, San Diego, 2016.

[6] A. Seaman, T.-s. Dao and J. McPhee, "A Survey of Mathematics-based Equivalent-circuit and Electrochemical Battery Models for Hybrid and Electric Vehicle Simulation," *Journal of Power Sources,* vol. 256, pp. 410-423, 2014.

[7] V. Ramadesigan, P. Northrop, S. De, S. Santhanagopalan, R. Braatz and V. Subramanian, "Modeling and Simulation of Lithium-ion Batteries from a Systems Engineering Perspective," *Journal of the Electrochemical Society,* vol. 159, no. 3, pp. R31-R45, 2012.

[8] T. Kim and W. Qaio, "A Hybrid Battery Model Capable of Capturing Dynamic Circuit Characteristics and Nonlinear Capacity Effects," *IEEE Transactions on Energy Conversion,* vol. 26, no. 4, pp. 1172-1180, 2011.

[9] D. Karthikeyan, G. Sikha and R. White, "Thermodynamic Model Development for Lithium Intercalation Electrodes," *Journal of Power Sources,* vol. 185, no. 2, pp. 1398-1407, 2008.

[10] Linear Technologies Corporation, "LTC2309 - 8-Channel, 12-Bit SAR ADC with I2C Interface," 2008. [Online]. Available: http://cds.linear.com/docs/en/datasheet/2309fd.pdf. [Accessed November 2016].

[11] J. Mathews and K. Fink, "Numerical Optimization," in *Numerical Methods Using Matlab, 4th Edition*, Upper Saddle River, Prentice Hall, Inc, 2008, pp. 430-436.

[12] M. Daigle, "Prognostics Algorithm Library [Computer Software]," 2016. [Online]. Available: https://github.com/nasa/PrognosticsAlgorithmLibrary. [Accessed November 2016].

[13] M. Daigle, "Prognostics Model Library," 2016. [Online]. Available: https://github.com/nasa/PrognosticsModelLibrary. [Accessed November 2016].

[14] J. J. Ely, S. V. Koppen, T. X. Nguyen, K. L. Dudley, G. N. Szatkowski, C. C. Quach, S. L. Vazquez, J. J. Mielnik, E. F. Hogge, B. L. Hill and T. H. Strom, "Radiated Emissions from a Remote-Controlled Airplane-Measured in a Reverberation Chamber," NASA, Hampton, 2011.

[15] S. Särkkä, Bayesian Filtering and Smoothing, New York: Cambridge University Press, 2013.

Appendix A

The purpose of this appendix is to give the reader a sense of the development of the Unscented Kalman Filter, UKF. Presented here is the basic general Gaussian filter with descriptions of both quadrature and cubature integration used in implementing the UKF [15]. While the provided explanations do not serve as mathematical proofs, they do give the reader a sense for how the various methods were developed along with applications.

The system model is assumed to be nonlinear and Markovian. Here we have that

$$x_k = f(x_{k-1}) + q_{k-1}$$

$$y_k = h(x_{k-1}) + r_k$$

where q_{k-1} and r_k are independent Gaussian noise. The variable *k* is the time step and $f(\cdot)$ and $h(\cdot)$ are nonlinear transforms. The measurement at *k* is y_k and the *n* dimensional state to be estimated is x_k. It is the goal of the Bayesian filter to return state estimates given past output measures. The state and measurement probability density functions may be expressed as

$$x_k \sim p(x_k|x_{k-1})$$

$$y_k \sim p(y_k|x_k)$$

The Bayesian filter estimates the state density given all past measurements up to *k*; that is estimate $p(x_k|y_{1:k})$. Per Bayes' rule, we may write the probability density function conditioned on $y_{1:k-1}$ as follows

$$p(x_k|y_k, y_{1:k-1}) = \frac{p(y_k|x_k, y_{1:k-1})p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}$$

Because of the Markovian assumption and application of the Chapman-Kolmogorov equation to the denominator, we may write

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{\int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k}$$

The above expression may be used in the Bayesian filter algorithm. The filter consists of three basic steps: initialization, prediction, and update. We start with some initial estimate of the state density to be estimated $p(x_0)$. Given measurements up to and including time step *k*-1, we can predict state x_k before measurement y_k with the Chapman-Kolmogorov equation

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}$$

To update the estimate at time step k, we use the measurement y_k and Bayes rule to evaluate $p(x_k|y_{1:k})$. In general, a closed form solution does not exist for the Bayesian filter. However, if both $f(\cdot)$ and $h(\cdot)$ are linear functions, the Kalman filter may be used.

To use the Bayesian filter for nonlinear state estimation, some assumptions and approximations must be made. First, we assume that the system output of the nonlinear transform is approximately Gaussian. In general, this is not the case; but the Gaussian approximation may provide a close enough fit to meet filtering requirements. There is also a need to solve integrals in real time that in general do not have closed form solutions. To do this, numerical solutions are used to approximate the integral. To lay out the filter algorithm, consider the following matrix relation for Gaussian random variables: $x \sim N(x|\mu, P)$ where *x* is a Gaussian vector with mean μ and covariance matrix *P*. The vector *x* can be partitioned into two sections.

$$x = \begin{bmatrix} x_a \\ x_b \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix} \quad P = \begin{bmatrix} P_a & P_c \\ P_c^T & P_b \end{bmatrix}$$

The conditional probabilities of the joint random variables x_a and x_b are given by

$$p(x_a|x_b) = N\left(x_a|\, \mu_a + P_c P_b^{-1}(x_b - \mu_b),\ P_a - P_c P_b^{-1} P_c^T\right)$$

$$p(x_b|x_a) = N(x_b|\, \mu_b + P_c^T P_a^{-1}(x_a - \mu_a),\ P_b - P_c^T P_a^{-1} P_c)$$

The above identity may be used to determine the Bayesian filter algorithm through moment matching. We can define the joint probability density function for x_k and y_k and next determine the conditional density of x_k given the new measure y_k has occurred in the update step. Using the templet above to stage our construction, we have the following.

$$p(x_k, y_k) = N(x|\mu, P) \qquad x = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad \mu = \begin{bmatrix} m_k^- \\ \mu_k \end{bmatrix} \quad P = \begin{bmatrix} P_k^- & C_k \\ C_k^T & S_k \end{bmatrix}$$

$$p(x_k \mid y_k) = N(x_k|m_k, P_k)$$

$$m_k = m_k^- + C_k S_k^{-1}(y_k - \mu_k) \qquad P_k = P_k^- - C_k S_k^{-1} C_k^T$$

The algorithm is shown below for the case of additive noise and nonlinear system model $f(\cdot)$ and measurement model $h(\cdot)$.

Algorithm for General Gaussian Filter

Prediction: We first need to predict the mean and covariance of the state x_k before a measurement occurs. We do this by using the state model along with our best estimate of the density function, assumed to be Gaussian. Of the mean and covariance, we have

$$m_k^- = \int f(x_{k-1}) N(x_{k-1}|m_{k-1}, P_{k-1}) dx_{k-1}$$

$$P_k^- = \int \{f(x_{k-1}) - m_k^-\}\{f(x_{k-1}) - m_k^-\}^T\, N(x_{k-1}|m_{k-1}, P_{k-1}) dx_{k-1} + Q_{k-1}$$

Update: To update our estimate of the state x_k given the recent measurement y_k, we use the conditional density relationship $p(x_k \mid y_k)$ as derived above using moment matching as well as the

definitions of mean and covariance for a random variable given its probability density function. We assume a Gaussian distribution. Although in general the output of a nonlinear system given a Gaussian input is not Gaussian, we make this assumption as an approximation to design a filter realizable in real time.

$$\mu_k = \int h(x_k) N(x_k |\, m_k^-\,,\, P_k^-) dx_k$$

$$S_k = \int \{h(x_k) - \mu_k\}\{h(x_k) - \mu_k\}^T N(x_k |\, m_k^-\,,\, P_k^-)\, dx_{k-1} + R_k$$

$$C_k = \int (x_k - m_k^-)\{h(x_k) - \mu_k\}^T N(x_k |\, m_k^-\,,\, P_k^-)\, dx_{k-1} + R_k$$

With solutions to the above integrals, we can finish the recursion using the results from moment matching.

$$m_k = m_k^- + C_k S_k^{-1}(y_k - \mu_k) \qquad P_k = P_k^- - C_k S_k^{-1} C_k^T$$

In general, closed form solutions to the integrals are not possible. Numerical integration techniques are used to approximate the integrals for real time implementation. Two techniques are considered here: Gauss-Hermite and Spherical cubature.

For Gauss-Hermite integration, consider the definition of the mean given that x is a Gaussian random vector and $g(\cdot)$ is some transform.

$$\int g(x) N(x|m,P) dx = \frac{1}{(2\pi)^{n/2}|P|^{1/2}} \int g(x) exp\left\{-\frac{1}{2}(x-m)^T P^{-1}(x-m)\right\} dx$$

To use Gauss-Hermite integration, we must use a change of variables to make the distribution zero mean and unit covariance. Let $x = m + \sqrt{P}\xi$ and recalling that we must take the determinant of the Jacobean for multivariable change of variables we have

$$\frac{1}{(2\pi)^{n/2}|P|^{1/2}} \int g\big(m + \sqrt{P}\xi\big) exp\left\{-\frac{1}{2}\big(m + \sqrt{P}\xi - m\big)^T P^{-1}\big(m + \sqrt{P}\xi - m\big)\right\} \big|\sqrt{P}\big| d\xi$$

Since the covariance matrix is positive semi-definite, we have $\big|\sqrt{P}\big| = |P|^{1/2}$. We now have the desired form.

$$\int g(x) N(x|m,P) dx = \int g\big(m + \sqrt{P}\xi\big) N(\xi|0,I) d\xi$$

One way to approximate this integral is to use Hermite polynomials. This approximation comes from the mathematics and physics communities and is simply stated here without derivation.

$$\int g(x) N(x|m,P) dx \approx \sum_{i_{1,\ldots,}i_n} W_{i_1,\ldots,i_n} g\big(m + \sqrt{P}\xi^{(i_1,\ldots i_n)}\big)$$

The sigma points ξ are the roots of the p^{th} order Hermite polynomial and the weights *W* are given by

$$W_{i_1,\dots i_n} = \frac{p!}{p^2\{H_{p-1}\xi^{i_1}\}^2} \times \dots \times \frac{p!}{p^2\{H_{p-1}\xi^{i_n}\}^2}$$

Since this integration method takes an *n* dimensional integral and breaks it down to *n* one dimensional integrals, the number of sigma points grows on order p^n.

To reduce the computational burden of numerical iteration, spherical cubature integration is used. The algorithm is presented here without development.

Consider the *2n*-point approximation to the multi-dimensional Gaussian integral

$$\int \mathbf{g}(\boldsymbol{\xi}) N(\boldsymbol{\xi} \mid \mathbf{0}, \boldsymbol{I}) d\boldsymbol{\xi} \approx W \sum_{i=1}^{2n} g\left(c\boldsymbol{u}^{(i)}\right)$$

Here *W* and *c* are scalar constants to be determined. There are *2n* symmetric vectors, each of length *n*. As an example, consider the case for $n = 3$.

$$u = \left\{u^{(1)}, u^{(2)}, u^{(3)}, u^{(4)}, u^{(5)}, u^{(6)}\right\} = \left\{\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}-1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\-1\end{bmatrix}\right\}$$

We can determine the constants W and c by considering two exact solutions; one for $g_j(\boldsymbol{\xi}) = 1$ and one for $g_j(\boldsymbol{\xi}) = \xi_j^2$. The subscript *j* here is for the j^{th} element of a given vector from the set of *2n* vector set *u*. For the case of $g_j(\boldsymbol{\xi}) = 1$ we have that

$$\int (1) N(\boldsymbol{\xi} \mid \mathbf{0}, \boldsymbol{I}) d\boldsymbol{\xi} = 1 = W \sum_{i=1}^{2n} 1 = W2n \qquad W = \frac{1}{2n}$$

For the case of $g_j(\boldsymbol{\xi}) = \xi_j^2$ we have that

$$\int \xi_j^2 N(\boldsymbol{\xi} \mid \mathbf{0}, \boldsymbol{I}) d\boldsymbol{\xi} = 1 = W \sum_{i=1}^{2n} \left\{c u_j^{(i)}\right\}^2 = W2c^2 \qquad c = \sqrt{n}$$

For example, consider the case of $n = 3$ as described above. If we choose the second element, for example, we can write

$$W \sum_{i=1}^{2n} \left\{c u_2^{(i)}\right\}^2 = W\{c^2 0^2 + c^2 1^2 + c^2 0^2 + c^2 0^2 + c^2 (-1)^2 + c^2 0^2\} = W2c^2 = \frac{1}{2n} 2c^2 \quad c = \sqrt{n}$$

As we did with Gauss-Hermite integration, we can do a change of variables to generalize the spherical cubature integration.

$$\int g(\boldsymbol{x})N(\boldsymbol{x}|\boldsymbol{m},\boldsymbol{P})dx \approx \frac{1}{2n}\sum_{i=1}^{2n} g\left(\boldsymbol{m}+\sqrt{\boldsymbol{P}}\boldsymbol{\xi}^{(i)}\right) \quad \boldsymbol{\xi}^{(i)} = \sqrt{n}\boldsymbol{u_i}$$

The approximation above is very like the unscented transform. If we include a zero-vector sigma point and an additional parameter for tuning, we can obtain the unscented transform.

By using either the Gauss-Hermite or spherical cubature integration to solve the integrals of the Kalman filter algorithm, one obtains a realizable implementation for nonlinear state estimation.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-06-2017	Technical Memorandum	

4. TITLE AND SUBTITLE

Implementation of a Battery Health Monitor and Vertical Lift Aircraft Testbed for the Application of an Electrochemisty-based State of Charge Estimator

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Potteiger, Timothy R.; Eure, Kenneth W.; Levenstein, David

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

533127.02.16.07.01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-20811

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA-TM-2017-219629

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified
Subject Category 06
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Prediction methods concerning remaining charge in lithium-ion batteries that power unmanned aerial vehicles are of critical concern for the safe fulfillment of mission objectives. In recent years, lithium-ion batteries have been the power source for both fixed wing and vertical lift electric vehicles. The purpose of this document is to describe in detail the implementation of a battery health monitor for estimating the state of charge of a lithium-ion battery and a lithium-ion polymer battery that is used to power a vertical lift aircraft test-bed. It will be demonstrated that an electro-chemistry based state of charge estimator effectively tracks battery discharge characteristics and may be employed as a useful tool in monitoring battery health.

15. SUBJECT TERMS

Battery health monitoring; Electrochemistry model; Nonlinear state estimation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	38	19b. TELEPHONE NUMBER *(Include area code)* (757) 864-9658